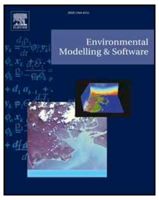
Environmental Modelling and Software 118 (2019) 120–133

Contents lists available at ScienceDirect

Environmental Modelling & Software

journal homepage: www.elsevier.com/locate/envsoft



A tile service-driven architecture for online climate analysis with an application to estimation of ocean carbon flux

Weiwen Ye[a,b], Feng Zhang[a,b], Yan Bai[c], Zhenhong Du[a,*], RenYi Liu[b]

[a] Institute of Geographic Information Science, Zhejiang University, Hangzhou, 310028, China
[b] Zhejiang Provincial Key Lab of Geographic Information Science, Hangzhou, 310028, China
[c] State Key Laboratory of Satellite Ocean Environment Dynamics, Second Institute of Oceanography, SOA, Hangzhou, 310012, China

ARTICLE INFO

Keywords:
Tile service
Virtual globe
High-efficiency online analysis
Climate change

ABSTRACT

The deteriorating environmental system and increasing number of time-varying datasets have generated a significant demand for new techniques to support online high-efficiency climate analysis. In response to this demand, this paper introduces an innovative multilayer architecture driven by a tile service in a virtual globe environment. This architecture is supported by the open-source and highly efficient osgEarth virtual globe. In addition, we develop a gSoap-based tile service to address the challenges of visualizing and analysing massive data under hardware restrictions conditions. The service contains a lossless pyramid tile set that can automatically provide tiles with proper accuracy according to the demands of users under different conditions. Furthermore, we add a hybrid database file system to the service for high availability. This service is then applied for estimation of ocean carbon flux. And a proof-of-concept prototype, SatCO2, has been developed to demonstrate the feasibility and performance of the proposed architecture.

1. Introduction

Since the late 19th century, the planet's average surface temperature has risen by approximately 0.9 degrees Celsius, a change driven largely by increased carbon dioxide and other human-made emissions into the atmosphere (NASA 2018). Although the warming of the planet will be gradual, the increasing frequency and severity of extreme weather events, like intense storms, heat waves, droughts, and floods, will be abrupt and acutely felt (Organization et al., 2008). These extreme events are interacting to generate emerging public health threats that endanger the health and well-being of hundreds of millions of people (Myers and Patz, 2009). Thus, understanding the basic structure of the climate system and seeking new techniques for diagnostic insight into climate changes has become an urgent research topic for climate scientists.

The virtual globe technique, which is a new data processing and analysis tool that can integrate heterogeneous geospatial data from real-time in-situ measurements, remote sensing (RS) and geographic information systems (GISs) at the global scale, has been utilized to support spatial analysis and decision making in climate change research (Yu and Gong, 2012; Liang et al., 2014; Chen and van Genderen 2008; Liu et al., 2015a). Since the world we are facing is three-dimensional, the spatial dynamic changes in climatic phenomena cannot be shown if we use traditional two-dimensional methods for visual display and simulation. As a tool to connect people and big data, the virtual globe technique addresses visible information directly and provides a real-time ability for interaction simulations.

As an example, for the past few years, geo-browsers (e.g., Google Earth, Microsoft Virtual Earth, NASA World Wind, and ESRI ArcGIS Explorer) have brought significant benefit to studies in earth system science, especially for climate change research (Gould et al., 2008; Gong et al., 2011). In the meantime, massive amounts of data have been accumulated under long-term observations, as satellite remote sensing has been widely used with its large-scale and real-time advantages. In addition, petabyte-scale archives have become freely available from multiple U.S. government agencies including NASA, NOAA and the U.S. Geological Survey (Woodcock et al., 2008; Vasco et al., 2010; Loveland and Dwyer, 2012; Nemani et al., 2011). That is, people can communicate the data downloaded from the abovementioned agencies and their research findings in an intuitive three-dimensional global perspective (Yu and Gong, 2012). However, the download action is essentially creating a copy from the hard disks of the server to the users and is sure to encounter bottlenecks. For example, it greatly raises the requirements for both the hardware and the network conditions of the

* Corresponding author. Institute of Geographic Information Science, Zhejiang University, Room 236, Main Teaching Building, Xixi Campus, Tianmushan Road 148, Hangzhou, ZheJiang Province, 310028, China.
E-mail address: duzhenhong@zju.edu.cn (Z. Du).

https://doi.org/10.1016/j.envsoft.2019.04.005
Received 27 April 2018; Received in revised form 12 November 2018; Accepted 5 April 2019
Available online 12 April 2019

Software availability

Name	Satellite-based marine carbon monitoring and analysis system (SatCO2)
Developer	State Key Laboratory of Satellite Ocean Environment Dynamics & Zhejiang Provincial Key Laboratory of Geographic Information System
Contact address	duzhenhong@zju.edu.cn
Year first available	2016
Hardware required	Discrete graphics, recommend NVIDIA
Program languages	C + +
Availability	http://www.satco2.com
Cost	Free (to researchers)

users when the data are truly immense. Moreover, in most cases, users may have to download the desired data and acquire specific visualization tools, which require specialized expertise and training (Zhang et al., 2016). As a result, for the convenience of climate researchers, the need to establish a method by taking full advantage of these resources is becoming more pressing.

In this paper, we introduce a tile service-driven architecture for high-efficiency visual simulation and analysis through the Internet. This study has two main objectives: (1) to realize the rapid display of massive data on a three-dimensional virtual globe under limited network bandwidth; and (2) to provide efficient statistical analysis for high concurrency requests under restricted hardware conditions. The remainder of this paper is structured as follows. Section 2 provides an overview of the works related to this paper. Section 3 introduces the architecture design, and Section 4 describes the methodology in detail. Section 5 evaluates the feasibility and efficiency of the architecture using a proof-of-concept prototype, and Section 6 concludes the paper and discusses future research directions.

2. Related works

In recent years, data visualization based on the virtual globe has played an increasingly important role in representing and analyzing scientific data (Standart et al., 2011). As an illustration, Li et al. (2011) adopted a systematic framework for visualizing dynamic geoscience phenomena in virtual globe environments. Liu et al. (2015b) proposed a systematic meteorological data visualization framework in World Wind to visualize and analyse meteorological data. Li and Wang (2017) introduced a new web-based platform for visualizing multidimensional, time-varying climate data on a virtual globe. Although these approaches work well for rendering large-scale geoscience phenomena, challenges still exist that require further research. As an example, the cases above are visualization systems that lack underlying support for statistical analysis applications. Moreover, as remote sensing data are strong in real time and have a complex structure, the increasing sizes of datasets often exceed the computational capabilities of local processors; thus, data computation problems need to be considered emphatically.

In another dimension, however, Google released Earth Engine for planetary-scale geospatial analysis; the images ingested into Earth Engine are pre-processed to facilitate fast and efficient data access for users. Additionally, to enable fast visualization during algorithm development, a pyramid of reduced-resolution tiles was created for each image and stored in the tile database (Gorelick et al., 2017). The image pyramid method was first introduced by Adelson et al. (1984) to improve the speed of real-time display and zoom of original images. An image pyramid is a collection of images, all arising from a single original image, that uses a quadtree structure to represent an image in which successively deeper levels represent successively finer subdivisions of the image area (Open C V, 2013; Hunter and Steiglitz, 1979). Similar to Earth Engine, the image pyramid concept was also applied to the visualization of large Mars (Powell et al., 2010) and lunar images (Chang et al., 2011). The main idea was to break images into workable sized tiles, process each tile in parallel, and finally merge the processed tiles to produce the requested result for visualization (Soille et al., 2018). Specifically, rather than processing the whole dataset, the process only calls the parts of images that users desire to participate in the computation, greatly enhancing the calculation efficiency. Although Earth Engine has enormous computing power and can visualize very large datasets, the display is two-dimensional and not vivid enough to effectively represent the dynamic change of three-dimensional or four-dimensional phenomena (Cheng et al., 2019).

To address these issues, we propose a virtual globe-based architecture driven by a tile service for rapid transmission, high-efficiency geospatial analysis and vivid visualization of vast amounts of data through the Internet. The proposed architecture allows investigators to explore digital information intuitively in a true three-dimensional environment. The next section describes the architecture design and key techniques.

3. Architecture design

Virtual globes are highly capable of supporting image, vector, and terrain data but have a limited capability for the online visualization and analysis of massive data (Liang et al., 2014). As shown in Fig. 1, an innovative multilayer architecture is proposed to visualize vast amounts of data for analysis through the Internet. Working from the bottom up, the architecture includes a tile service layer, a 3D engine layer and a data engine layer.

The tile service layer, the core of the architecture, is the tile data source interface that is composed of a tile set, a hybrid database file system and a web service. This tile set is organized by an image pyramid structure in a lossless way. The tiles are then compressed and stored in Hadoops distributed file systems (HDFS; Fortner, 1998). During data processing, the key statistics and the metadata are inserted into a PostgreSQL database for faster future retrieval and calculation. Additionally, the database is compatible with PostGIS for heterogeneous data applications, such as in-situ data. The above components make up a hybrid database file system. A detailed description of this method can be found in Section 4.1. Above this layer is the 3D engine layer, which provides a platform for high-performance three-dimensional

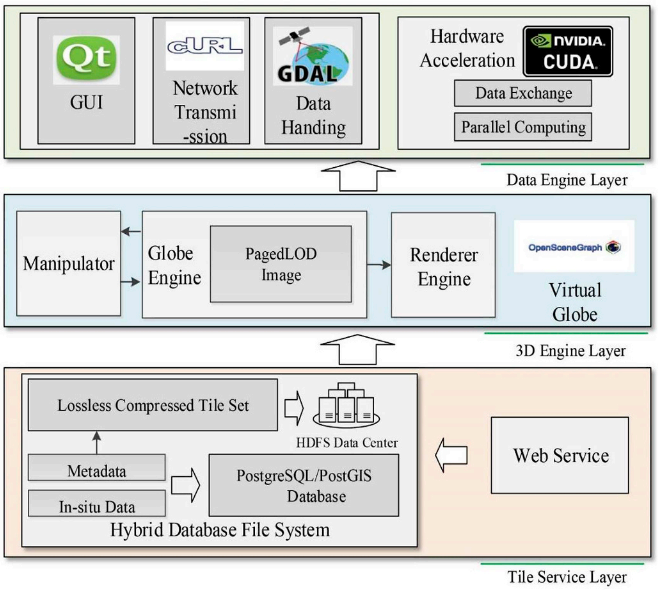


Fig. 1. The multilayer architecture design. The architecture is driven by a tile service in a virtual globe environment to support online high-efficiency climate analysis.

visualization.

The 3D engine layer is implemented on top of osgEarth, which is an open source 3D GIS system that enables access to local and Internet-based raster and vector data sources (Liang et al., 2015). As shown in Fig. 2, the framework of osgEarth contains a core module, globe engine, which implements the data loading by a paged level of detail (LOD) quadtree structure. PagedLOD is derived from LOD and adds the ability to defer the loading of child nodes until they are actually required. For rapid response to the activation of the globe engine, the manipulator is composed of several handlers. As soon as the globe engine is activated, it will call the render engine to modify the scene graph by cull or draw traversals. Notably, the renderer engine is able to link and execute external GPU code, such as compute unified device architecture (CUDA) kernels and custom shaders.

Extending the osgEarth virtual globe, we integrate a plugin for receiving and processing compressed tiles. The plugin is available through cURL and GDAL (Geospatial Data Abstraction Library). As an open-source file-transmission tool, cURL follows URL syntax and works under the command line mode. The client sends data requests to the server and adopts cURL to decompress the data and then shows the data in real time on the osgEarth virtual globe after the dataset is read by GDAL. In addition, the cross-platform Qt provides a set of UI elements for classic desktop-style user interfaces. These libraries, together with a GPU-accelerated library for image parallel computing, form the data engine layer.

All of these layers work together to provide online large-volume data visualization and analysis applications in climate change research. Section 4 describes these mechanisms.

4. Methodology

As the multi-layers of images and geo-referenced visualization over an entire globe require a great amount of well-processed data which would be very difficult to create and manage in a stand-alone system (Wang et al., 2013), we develop a client-server like tile service for massive data analysis and visualization under online hardware restrictions and high concurrency request conditions.

4.1. Tile service for high-efficiency analysis

4.1.1. Data preprocessing and organization workflow

To realize online high-efficiency visualization and analysis, we adopt a pre-processing method for raw data to build a tile set based on the image pyramid structure for each image according to its spatial extent and spatial reference. Then, the client requests corresponding level tiles in accordance with the distance from the viewpoint for display; in the case of a statistical calculation, the system calls the highest-level tiles (lossless level tiles) to perform the computation.

This method focuses on resolving three problems:

1) Tile indexing problem. To reduce the tile retrieval time, the tiles need to be organized in an efficient and generic manner.
2) Tile accuracy problem. Ensure that the highest-level tiles are a full backup for the original image and that the backup procedure should be reversible.
3) Data redundancy problem. In situations of limited disk space and constantly updating data, the tile should be set to minimize the data size to reduce the pressure on the server's hard drive and to facilitate rapid transmission through the Internet.

We designed the following solution for accelerating the calculating and loading speeds of images without any loss in data accuracy to ensure real-time Internet-based analysis and visualization.

First, the tile indexing problem is inspired and addressed by the data loading method of osgEarth. As shown in Fig. 3, based on a pyramid model, osgEarth selects tiles to perform dynamic loading of large datasets. In other words, by recording the spatial reference and spatial extent information of the original image, osgEarth adopts a real-time image decomposition process by a TileKey technology that is the index of each tile and calculated by the distance from the viewpoint to conduct data management. TileKey is in a structure of (Level,X,Y). At level 0, the whole globe is divided into two hemisphere tiles, and each tile is managed by TileKey((0,0,0) and (0,1,0)). Then, as the distance from the viewpoint becomes closer, osgEarth continues to index the tiles with a TileKey from the lower levels to the higher levels according to a quadtree data structure. For every two adjacent levels, each tile in the upper level is equally divided into four lower-level tiles. As the levels go deeper, the tiles decrease until the spatial resolution is finer than that of the original image. With the refinement of the tiles, the range of each tile decreases gradually such that when doing data loading the corresponding tile will be loaded only if there are valid data from the original image range. This data loading method can improve the rendering efficiency by avoiding unnecessary data loading. But data processing also creates a certain efficiency loss.

Thus, the original image are preprocessed to tiles to avoid such efficiency loss. Similarly, the tiles are organized by the rule of TileKey, which means that the retrieving path of each tile file can be expressed by the folder (Level), subfolder (X), and tile file name (Y). We also regard the image file name as a unique identifier to reduce the tile retrieval time. Meanwhile, in the process of data preprocessing, the maximum pyramid level, projection coordinates, row number, column number and spatial extent of the original image are recorded as metadata for further usage.

According to the built-in TileKey, the osgEarth virtual globe can, therefore, perform direct rendering by resolving the corresponding tile paths to conduct real-time requests, which can be expressed as:

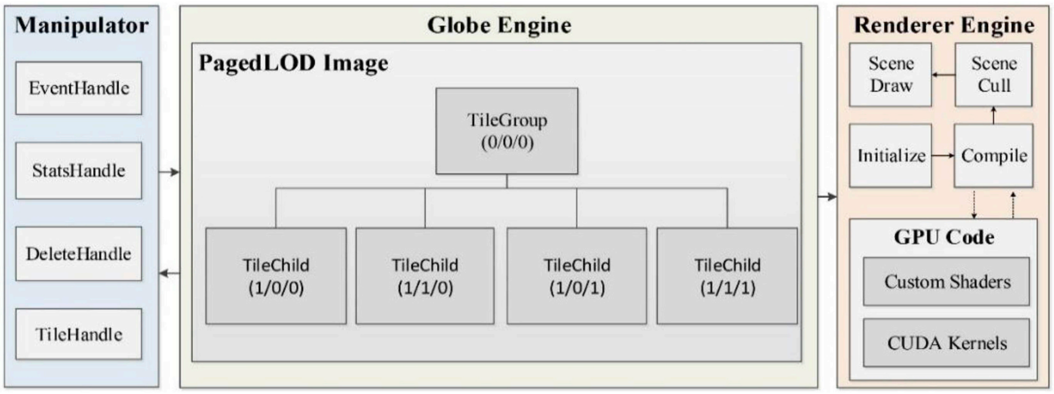


Fig. 2. Structure of the osgEarth virtual globe.

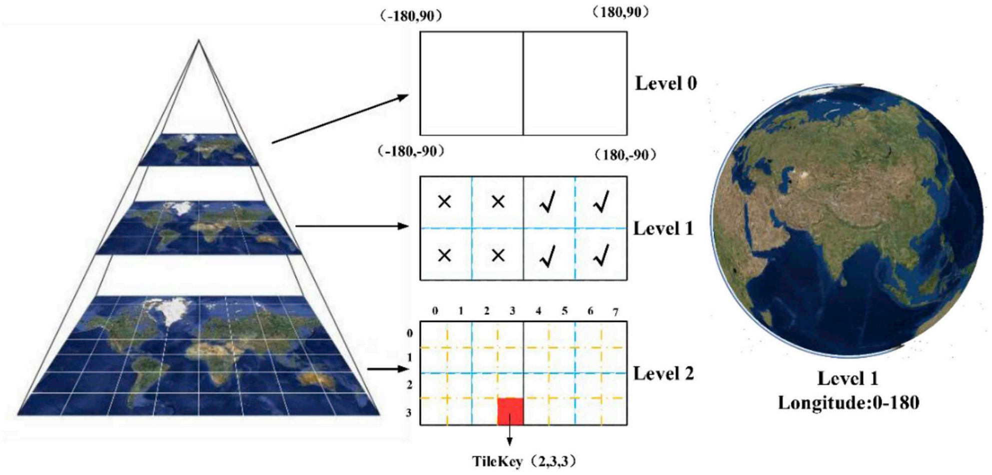


Fig. 3. Data management method of osgEarth. The original image is divided into tiles and adopts the pyramid method for dynamic loading. Each tile is indexed by a TileKey and tiles will be loaded only if there are valid data from the original image range.

$$f(\text{Level}, X, Y) = \text{http: \\\\ServerIP Address\\FileName\\Level\\X\\Y} \tag{1}$$

Second, there is a tile accuracy problem. To retain the pixel value of the original image, for any original image pixel (m, n) with a value α, we must find one pixel (p, q) among the tiles that has the same value α, and such mapping should be a one-to-one correspondence. Since osgEarth continues to decompose tiles from up to down until the spatial resolution of the tiles is finer than that of the original image, the pixel number of the maximum level tiles is larger than that of the original image. We set the maximum level that the original image can achieve as *Max* and regard all of the pixels among the tiles at level *Max* as set O. All the pixels of the maximum level tiles in the intersection of the same spatial extent of the original image are contained in set M, which is a subset of set O, with a pixel value set N. Similarly, all of the pixels of the original image are set P with a pixel value set Q. Since set M should be larger than set P, we define a pixel set R with a pixel value set S that is a subset of set M, and the mapping between set P and set R is a one-to-one correspondence. Fig. 4 shows the relationship description of all these sets. Thus, the main task would be the construction of a one-to-one correspondence mapping from set P to set R. Once this mapping relationship is generated, it is further processed to obtain the rest of the pixel values in set O/R (stand for set O minus set R). The method we adopt here is the nearest neighbour search method because it can retain all pixel values of the original image and is more efficient without losing accuracy. Extending the nearest neighbour search method, we integrate some modifications to further enhance the analysis efficiency.

The details of this method are as follows. We first illustrate the mapping of points from set P to set R (Fig. 5-a). To maintain a one-to-one correspondence in the mapping between set P and set R, we define the pixel in set R as having a coordinate $A_{Max,i,j}$ with a pixel value $\alpha_{Max,i,j}$; similarly, a pixel of set P can be denoted as $B_{l,m}$ with a pixel value $\beta_{l,m}$. The mapping between these two sets can be represented as:

$$f{:}B_{l,m} \sim A_{Max,i,j} \tag{2}$$

where i= < θl > , j= < γm> and <χ> indicates the round of χ. The two parameters γ and θ are the ratio of the row/column number from set M to set P. With this mapping function, we can find for each pixel in set P a unique pixel in set R, and we define this pixel value as $\alpha_{Max,i,j} = \beta_{l,m}$. However, there are still some pixels in set M that have not been valued that we call set M/R. Similarly, we construct a mapping from set M/R to set P:

$$g{:}A_{Max,i,j} \sim B_{l,m} \tag{3}$$

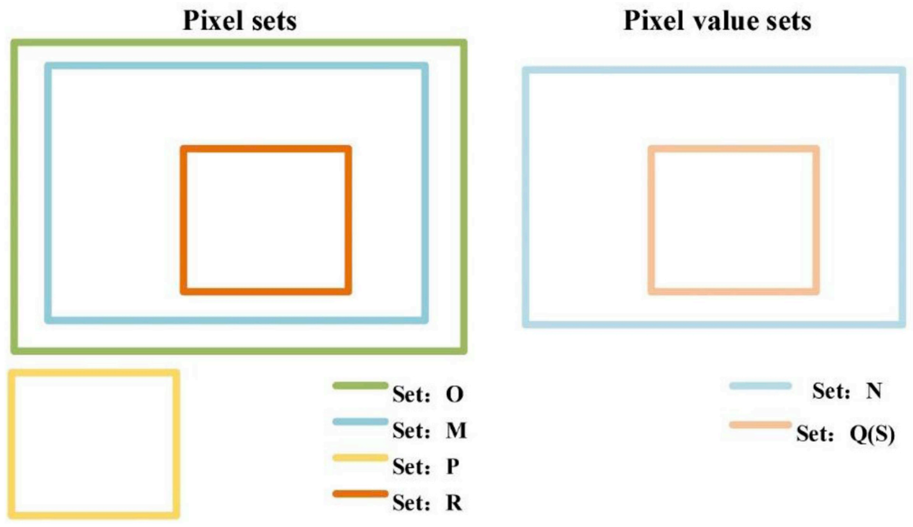


Fig. 4. Relationship illustration of all the pixel (values) sets.

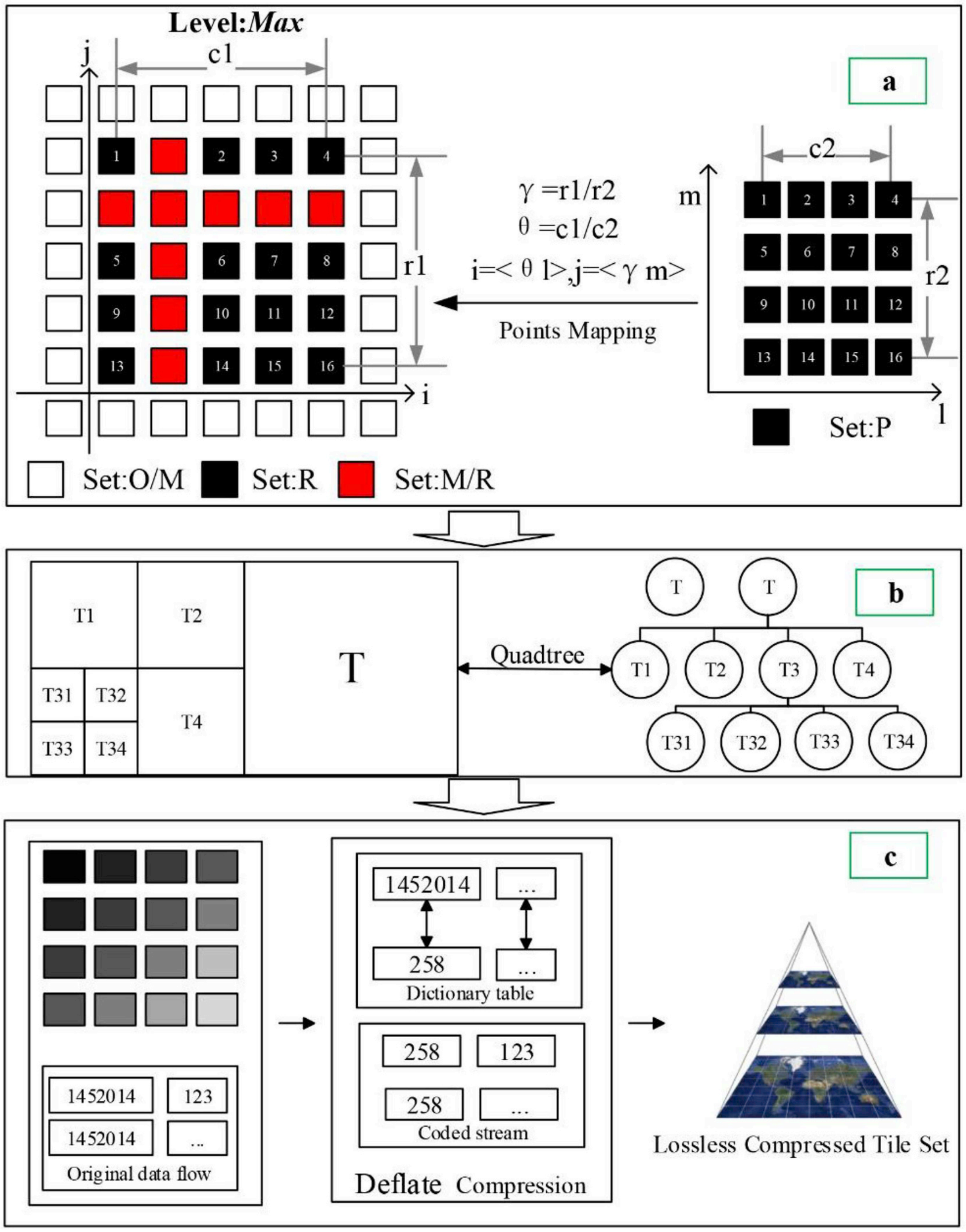


Fig. 5. Data processing workflow of the proposed method.

where l= $<\frac{i}{\theta}>$, m= $<\frac{j}{\gamma}>$. We add one more digit for all of the elements from set M/R to facilitate the restoration of the original image from the tiles, e.g., we assign 0.34321 to the pixel from set M/R when the corresponding original pixel value is 0.3432. Furthermore, by assigning NODATA to set O/M, we then obtain a 1:1 tile backup for the original image quickly and easily. Next, we adopt the quadtree data structure to obtain a pyramid tile set (Fig. 5-b).

Finally, we address the problem of data redundancy. As the tile set is already a 1:1 backup of the original image, we only need to save this tile set without the original image. Considering the data volume as well as its rapid growth, we designed the following compression strategy to reduce the consumption of disk space (Fig. 5-c). Because some parts of the original image are useless for our research, e.g., the parts blocked by overcast conditions, we set the values of these parts as NODATA (all of the NODATA values are rendered transparent on the virtual globe). Apart from NODATA, there are pixels with a float format that have poor predictability. Furthermore, although the compression strategy can reduce the disk footprint and improve the online file transmission speed, it can also result in decompression problems. In this situation, we adopt the Deflate (Deutsch, 1996) encoder as compression encoding and the cURL module as decompression engine for data requests. Deflate is a popular dictionary-based compression method that was originally used in the well-known Zip and Gzip software and has since

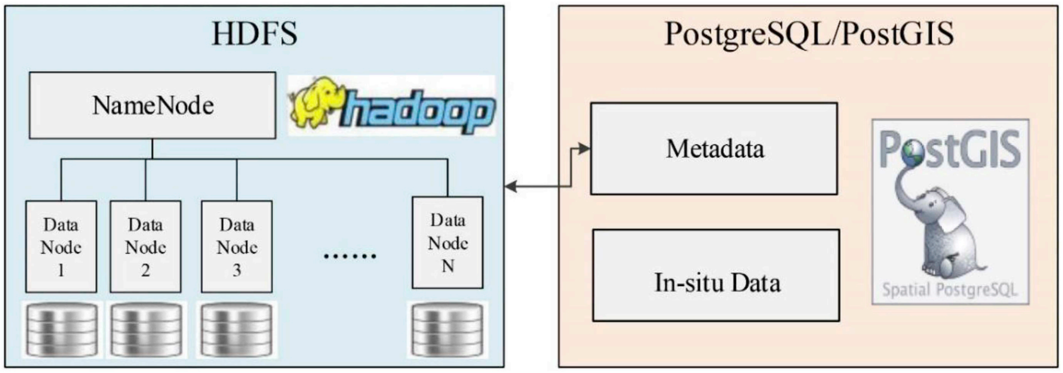


Fig. 6. The hybrid database file system design for heterogeneous data organization.

been adopted by many applications; the most important is the HTTP protocol (David, 2004). It operates by constructing a dictionary to register a string that appears repeatedly in the original data flow as a shorter string. Meanwhile, the cURL module supports Gzip and inflates content-encoding and performs automatic decompression.

As the tile set evolves, the requested tiles are transmitted from the server to the client. Upon decompression, the highest-level tiles go through a data restoration phase to reconstruct the original image part. Next, either the lower-level tiles or the restored image part are pushed to the GPU for rendering.

4.1.2. *Hybrid database file system for heterogeneous data organization*

On the server side, a large-volume tile set with its metadata are constructed. In addition, in-situ data are also needed in climate change research. All of these heterogeneous data should be unified into a data platform for organization purposes to shift the burden from the client to the server.

To accomplish this, a hybrid database file system is adopted for the unified organization of multi-source data and Fig. 6 shows the schema design. First, we adopt the HDFS to manage the tile set. The HDFS cluster consists of a single NameNode, a master server that manages the file system namespace and regulates access to files by clients. In addition, there are a number of DataNodes, which connect to the NameNode and respond to requests from the NameNode for filesystem operations (Borthakur, 2008). Hadoop, due to its scalability in storage and computing power, is a suitable platform for increasing volumes of remote sensing data (Lee and Lee, 2013). However, the HDFS has some deficiencies in the management of multisource data. For example, the smallest unit of vector data is a “record”, that is, specific time and space parameter information, and multiple records constitute a vector data. This structured way of organizing data can be disastrous for HDFS. To remedy this, in our hybrid database file system, the in-situ data and the metadata of original images are managed by PostgreSQL, while the spatial objects are stored in PostGIS. The PostgreSQL/PostGIS strategy not only reduces the retrieval pressure of the data node but also improves the query efficiency. Thus, by taking advantage of the spatial information service of PostgreSQL/PostGIS, the proposed hybrid database file system solves the problem of heterogeneous data organization and provides efficient data storage and access services.

4.1.3. *GSoap-based web service*

For the purpose of simplifying client-to-server interactions, we construct a web service based on gSOAP tools. As the heterogeneous data have been stored in the hybrid database file system, we need only to interact with the virtual globe, send the request to the server through the soap protocol and wait for the response after it finishes all of the computations based on the lossless tile set.

SOAP (Foo and Lee, 2002) is a lightweight protocol for exchanging structured information in a decentralized and distributed environment. The majority of C++ web service toolkit libraries provide a set of API functions to handle specific SOAP data models; however, users have to change their program structures to accommodate the relevant libraries. In contrast, gSOAP provides a kind of transparent SOAP API function by compiler technology, and the detailed contents of SOAP that are not related to the development are hidden to users. Thus, we need only to define a series of relevant APIs both in the client and in the server, thereby organizing the relevant parameters into XML messages. For multi-request cases such as time series queries, we combine a plurality of XML messages into one XML message based on the XML structural features. Upon receiving the message, the server parses the message and returns the result to the client. Similarly, the server also combines multiple results into one.

Taking temporal average along sections for example, we organize the XML keys by region of interest (ROI), file of interest (FOI) and MaxLevel. ROI contains the latitude and longitude information of each sections, FOI comprises the dataset type and start/end date of the composition period of satellite data information, and MaxLevel is the maximum pyramid level reached by the original image. All of this information can be easily accessed from the hybrid database. As an example, Table 1 is an XML encoding request of temporal average along sections query.

4.2. *Tile service based ocean carbon flux estimation*

For analysis purposes, with the proposed tile service, we can perform online high-efficiency ocean carbon flux estimation by large-sized remote sensing images. In this study, we adopt a commonly used method to calculate the air-sea CO2 flux for ocean carbon flux

Table 1
An XML case of temporal average along sections.

Send	Receive
< TemporalAverage >	< Max > 11,12,13 < /Max >
< ROI >	< Min > 6,7,8 < /Min >
< LONLAT1 > 123,45 < /LONLAT1 >	< Dev > 1,2,1 < /Dev >
< LONLAT2 > 124,46 < /LONLAT2 >	< Mean > 2,1,1 < /Mean >
< LONLAT3 > 127,47 < /LONLAT3 >	
< /ROI >	
< FOI >	
< DatasetType > EAMS < / DatasetType >	
< TimeSeries >	
< T1 > 20150601TO20150630 < /T1 >	
< T2 > 20150701TO20150731 < /T2 >	
< T3 > 20150801TO20150831 < /T3 >	
< /TimeSeries >	
< /FOI >	
< MaxLevel > 5 < /MaxLevel >	
< /TemporalAverage >	

estimation, which is based on the multiplication of the CO2 partial pressure difference between the surface seawater and the atmosphere and the CO2 gas transfer velocity. The equation is as follows:

$$\text{Flux} = \Delta p\text{CO2} \times \text{E} = (\text{WCP} - \text{ACP}) \times \text{K}_\text{H}^\text{CO2} \times \rho \times \text{C}_2 \times \text{k} \times 24 \times 10^{-2} \tag{4}$$

where WCP is the partial pressure of CO2 in seawater, ACP is the partial pressure of CO2 in atmosphere, $\text{K}_\text{H}^\text{CO2}$ is the dissolution efficient of CO2, ρ is the sea water density, C_2 is the wind speed coefficient and k is the gas transfer velocity. All of these unknown variables can be calculated from sea surface temperature (SST), sea surface salinity (SSS) and sea surface wind speed (SSW); the details are attached in Appendix-A.

These five remote sensing image data(WCP、ACP、SST、SSW and SSS) have been pre-processed and constantly updated in the hybrid database file system for high-efficiency analysis through the tile service. As the original image has been broken into lightweight workable tiles, the entire calculation process is applied to each tile in parallel (Fig. 7), and the calculation results can be loaded onto the virtual globe directly tile by tile without any merging. Similarly, the calculated flux tile data is stored in the hybrid database file system and can be used for external carbon budget estimation through the tile service. As mentioned above, users interact with the virtual globe, send the request to the server and wait for a response. Moreover, the estimation process only calls the parts of image that users desire to participate in the computation, greatly enhancing the calculation efficiency.

4.3. GPU-based volume rendering of remote tile sets

For visualization purposes, based on the tile service, a direct volume rendering method (Du et al., 2015) is adopted to realize the spatio-temporal visualization of remote time-varying air-sea CO2 flux tile sets. Volume rendering is a technique for rendering images of volumes containing a mixture of materials and can represent both the interiors and the boundaries between different regions (Drebin et al., 1988; Laur and Hanrahan, 1991). This technology has been widely used for the simulation of climate phenomena and the visualization of massive multidimensional datasets (Arthur et al., 2010; Darles et al., 2011; Song et al., 2006). Furthermore, with hardware advances, graphics processing units (GPUs) have been widely used to accelerate volumetric rendering for interactive applications (Monte et al., 2013).

Specifically, in this section, volume data reconstruction and updating are realized by CUDA parallel computing with the tile set on the server side, and a half-angle slice rendering technique (Green, 2008) is adopted for the dynamic visualization of climate data with a shadow effect. CUDA is a parallel computing platform and programming model developed by NVIDIA for general computing on graphical processing units (GPUs) (Nvidia, 2010). With CUDA, we are able to dramatically speed up the half-angle slice computing by harnessing the power of GPUs. The entire rendering process can be summarized as follows:

1) Generate or update the location and colour of the flux particles.
2) Send visualization results over the network from servers.
3) Calculate the half-angle array and order the particles along the array.
4) For each tile, select a set of particles for projection and render them from the viewpoint of both the viewer and the light.
5) Compose tiles to form the final image and repeat the rendering process.

Fig. 8 shows the effect of this half-angle volume rendering approach for air-sea CO2 flux dynamic simulation.

5. Demonstration and evaluation

5.1. System demonstration

Integrating the architecture we proposed above, the Satellite-based marine carbon monitoring and analysis system (SatCO2) is developed for ocean carbon flux estimations, and its database provides relevant remote sensing data.

By providing information on the influence of climatic variability

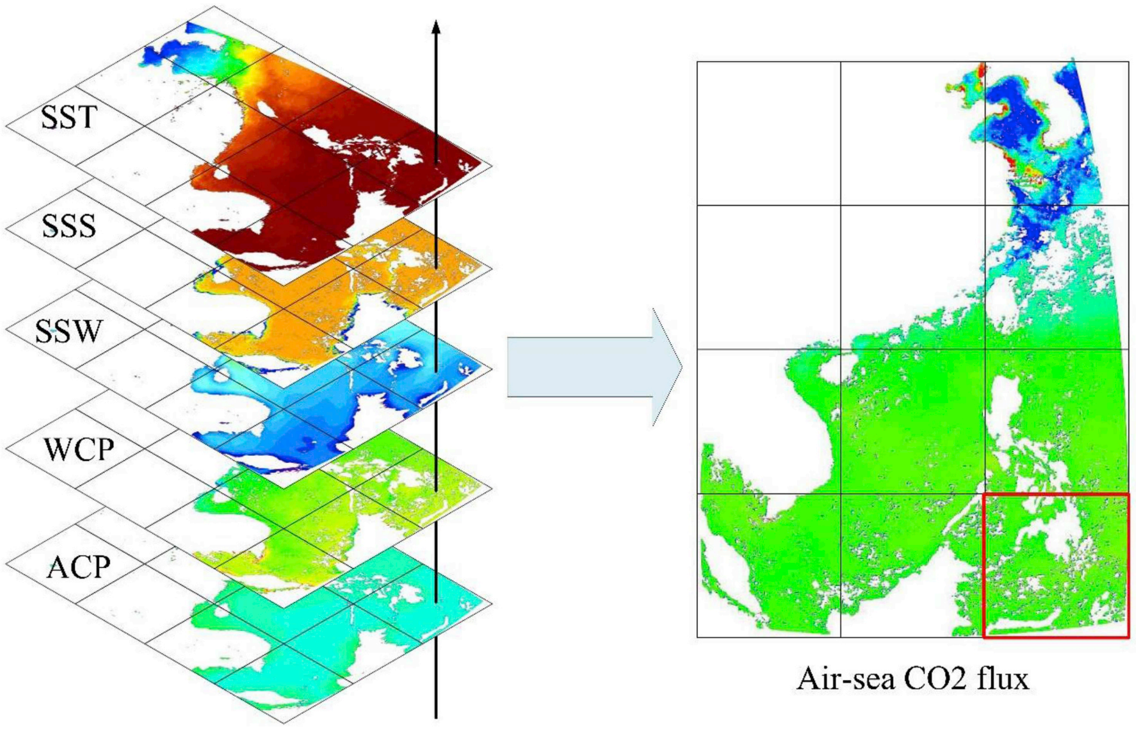


Fig. 7. An example of air-sea CO2 flux calculation by multiple remote sensing image data. The whole calculation process is applied to each tile in parallel.

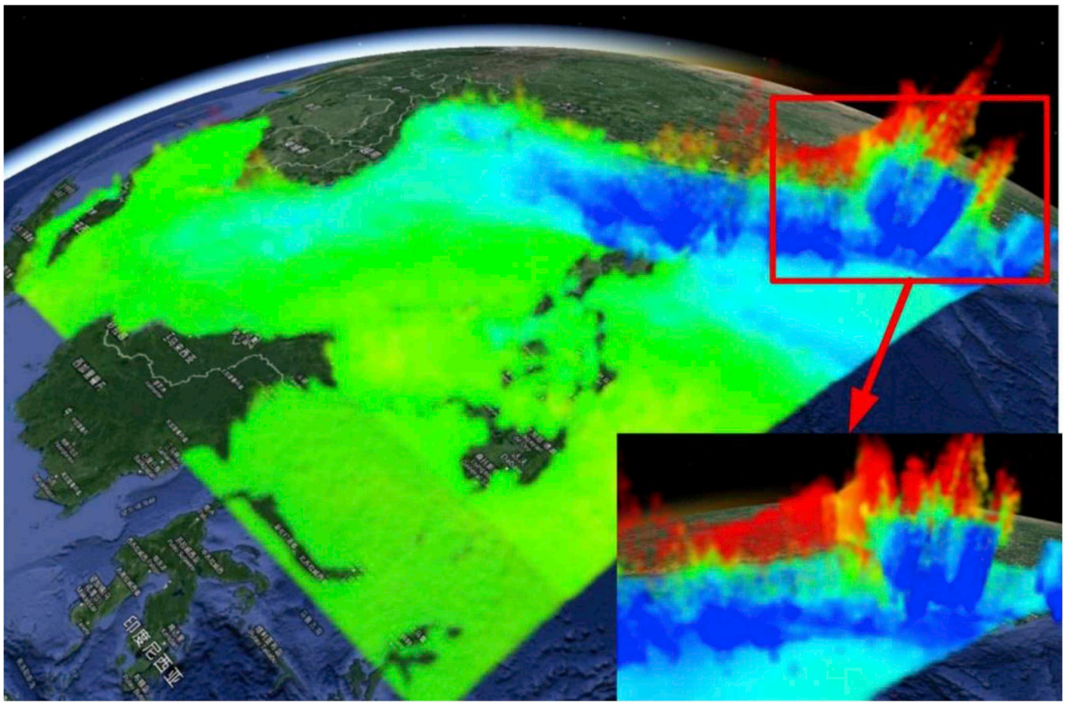


Fig. 8. Air-sea CO2 flux dynamic simulation by half-angle volume rendering. High emissions are represented by red, high absorption is represented by blue, and the height reached by particles represents the current carbon budget in the region. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

and change on carbon stocks and fluxes (Kang et al., 2014), Fig. 9 shows some screenshots of the SatCO2 interface. Fig. 9-a shows a view of monthly averaged sea surface wind data for May 2016, where blue represents a low wind speed, red represents a high wind speed, and the black vectors represent wind directions. As shown in Fig. 9-b, SatCO2 adopts the method mentioned in Section 4.2 to calculate the air-sea CO2 flux for ocean carbon flux estimation; Fig. 9-c shows the carbon budget calculation result of the East China Sea during April 2000. The air-sea carbon budget during April 2000 is −12.62 tC/yr.

5.2. Experiment evaluation

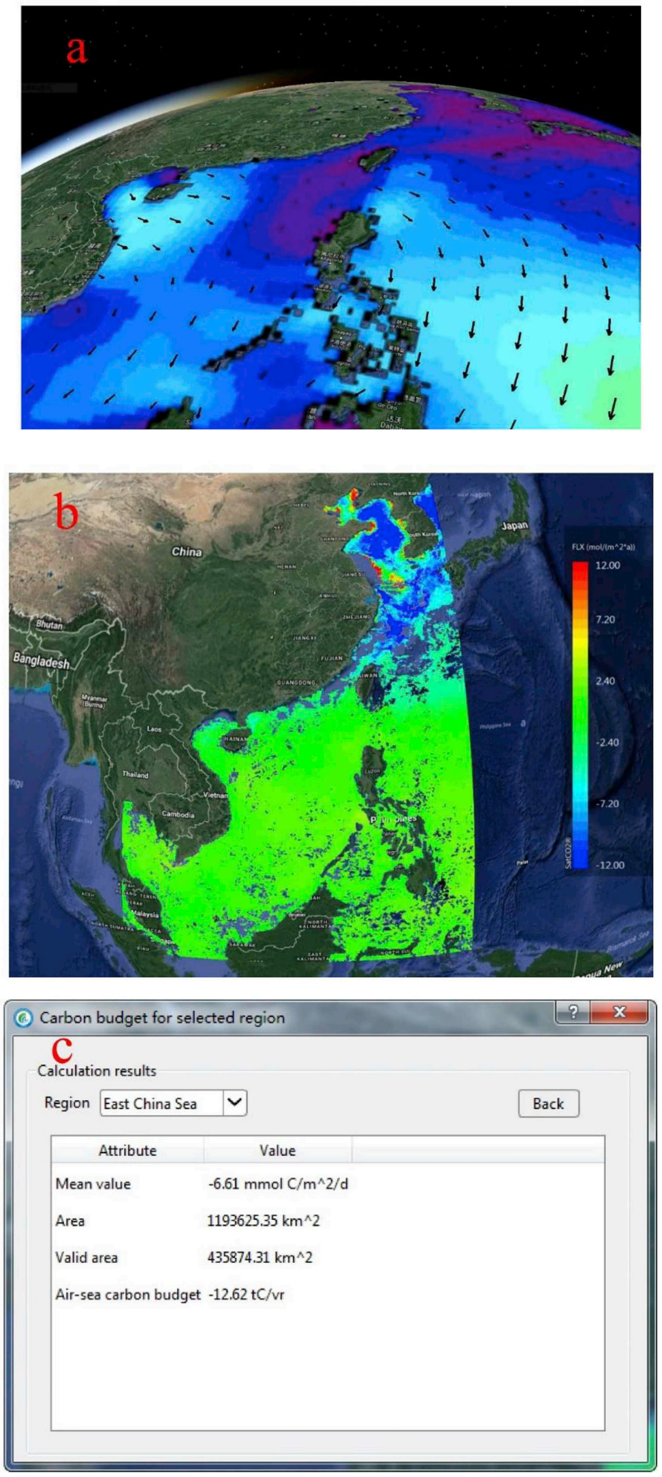
The data we use in this section for the experiments are ten-day average chlorophyll α concentration data from 1 January 2010 to 31 December 2015 in the South China Sea area (98°-127°E, 0–25°N) with a horizontal resolution of 2 km.

Section 5.2.1 compares the loading times of lossless tiles and the original image under different network bandwidths. We then compare the calculation times of time series data in different time spans under conditions of whether we adopt a lossless tile set in Section 5.2.2, and Section 5.2.3 shows the rendering performance evaluation by the spatiotemporal visualization process.

5.2.1. Enhanced efficiency of the data loading

The image animation module of SatCO2 (i.e., query and select the time series data from the database for animation, after which the selected data will be loaded to the virtual globe as layers) is applied to compare the loading times between the lossless tiles and the original image. To eliminate the uncertainty caused by data compression in our experiments, the original image is pre-processed with the same compression method mentioned in Section 4.1. We also repeat the experiment five times (the cache is cleared after each experiment) and choose the average result to avoid uncertainty factors such as bandwidth fluctuations as much as possible. Apart from this uncertainty, because of the inconsistent amount of tiles among different levels in the pyramid structure, only tiles from the maximum level are loaded to ensure that the detail of lossless tiles is the same as the original image. The experiment consists of five different bandwidths (4 Mbps, 10 Mbps, 20 Mbps, 50 Mbps and 100 Mbps) and ten variable data counts. The loading time costs and comparison results are shown in Fig. 10.

Fig. 10-a shows a comparison of the loading times among the original compressed images in different network bandwidths, while Fig. 10-b compares the lossless compressed tiles with the same parameter used in Fig. 10-a. Fig. 10-c shows the ratio of loading times between the original image in Fig. 10-a and the lossless compressed tiles in Fig. 10-b that are under the same bandwidth and data count. As we can see from Fig. 10-a, the relation between the loading time and the number of loading images is almost linear at a lower bandwidth (e.g., 10 Mbps and 4 Mbps), which means that the network bandwidth is the dominant factor. In Fig. 10-b, almost all of the loading times fluctuate between 4 and 5 s whenever the bandwidth changes. We can conclude that hardware conditions such as fluctuations in the memory could dominate in this case. It can be determined from Fig. 10-c that the advantage of the tiling strategy in data loading is quite obvious at low bandwidths compared with the time cost of the original image. However, this gap is narrowed as the bandwidth increases, and the loading time is almost the same when the bandwidth is broad enough, e.g., 100 Mbps in this experiment. We only simulate in the way of point-to-point transmission; if we are facing real-world applications insomuch that the upload bandwidths of servers and the download bandwidths of clients are both limited, it will challenge the transmission of original images substantially. In contrast, the tile strategy could behave better at speeding loading times up for the transmission of massive data from the server to the client under a limited network bandwidth. This phenomenon is mainly due to the characteristics of the architecture. It is able to do real-time operations and realize the block-by-block loading of pyramid tiles on demand, while the default method needs to wait for the entire original dataset to finish transmission.



Carbon budget for selected region

Calculation results

Region East China Sea

Back

Attribute	Value
Mean value	-6.61 mmol C/m^2/d
Area	1193625.35 km^2
Valid area	435874.31 km^2
Air-sea carbon budget	-12.62 tC/yr

Fig. 9. Screenshots of SatCO2. (a) A view of the monthly averaged sea surface wind data for May 2016, (b) the calculation result of air-sea CO2 flux in April 2000, (c) the carbon budget calculation result of the East China Sea during April 2000.

5.2.2. *Efficiency and accuracy promotion in geospatial analysis*

In traditional Euclidean geometry, both lines and polygons are composed of infinite points, but remote sensing images are made up of only finite pixels because of the concept of resolution. Based on the point level, the following experiment is carried out to compare the calculation efficiency and accuracy between the lossless tiles and the original image. Moreover, as multiple requests are combined into an entire XML message when performing a time series query, the independent variable is the data source. This experiment is performed on the server side to eliminate any consideration of network conditions, and to better demonstrate the results, all data are prepared and hosted in the server's file system.

By generating a random coordinate set in the original image, we select time series of different time spans and conduct a point time series query; then, the query result is saved to compare the accuracy. The time cost comparison is shown in Fig. 11.

As we can see from Fig. 11, the lossless tiling strategy can obviously improve the efficiency of the statistical analysis, and this enhancement is more remarkable as the analysis becomes more complex. The loss due to mapping from the original coordinates to TileKey is negligible, as the time complexity of the nearest neighbour search algorithm is O(n). Additionally, calling the tile of an image can significantly reduce the memory consumption and disk read-time. Thus, we pay more attention to the accuracy problem. Theoretically, the result of these two methods should be the same as the tile set is a full backup of the original data, and we can prove this to be true in our experiment by introducing the average error of the result as:

$$\text{RAE} = \frac{\sum |a_i - b_i|}{\sum b_i} \tag{5}$$

where b_i is the original value and a_i stands for the tile value. As shown in Table 2, the RAE from January 2010 to December 2010 is 0.

Based on these results, we can conclude that the tiling strategy can be much more efficient at statistical analysis than the original image with no loss of accuracy.

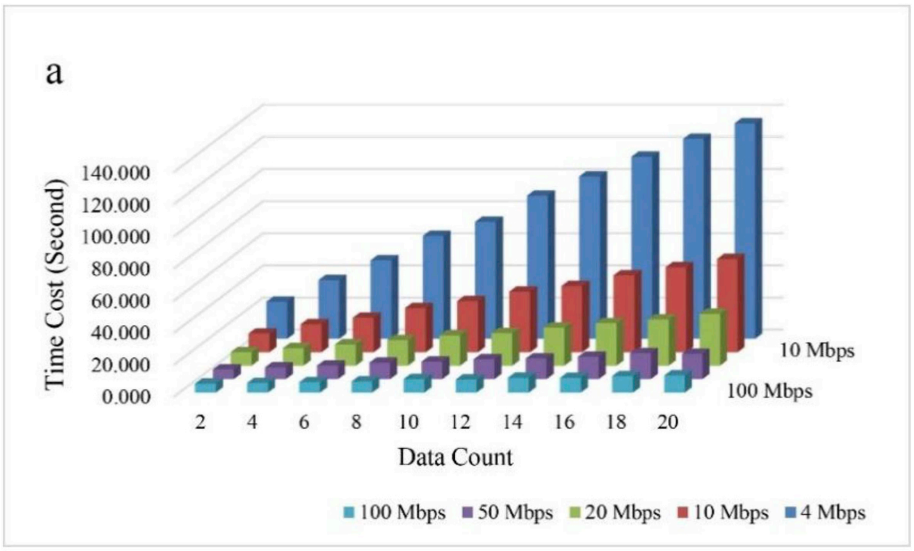
5.2.3. *Rendering performance evaluation*

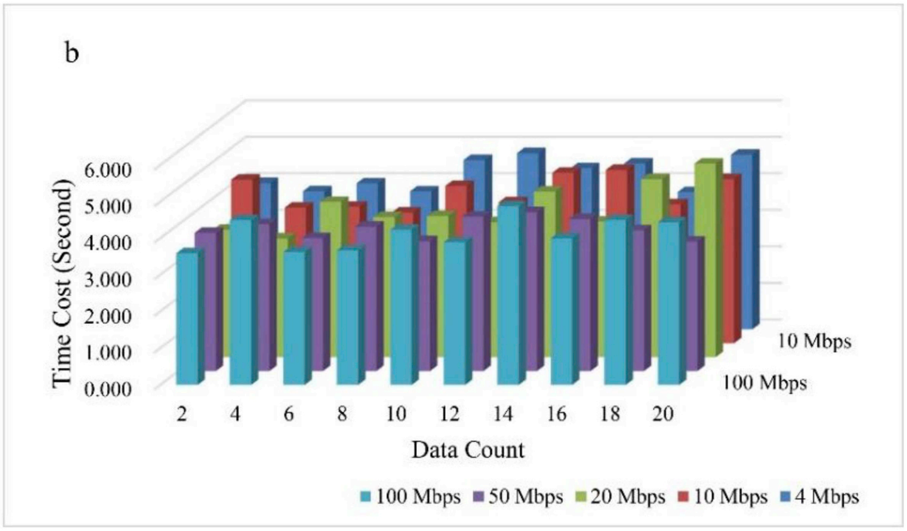
In this section, the experiment is performed on a PC with an NVIDIA GeForce GTX 660 2 GB graphics card and a 10 Mbps network. Fig. 12 shows the FPS with a spatiotemporal visualization of the different numbers of carbon flux tile data in a local area network. The minimum/maximum baseline is the rendering efficiency of Du's experiment (2015) in a standalone PC with the same graphics card. The result shows that the average FPS of the tile volume rendering exceeds the baseline. This is because the tiles can be rendered as they reach the client rather than having to wait for the original image to be read from the disk into memory before the data reconstruction and update processing can begin. Thus, we can conclude that tile service technology improves online rendering performance dramatically.

6. Conclusions and discussion

The objective of this paper is to propose a systematic architecture for online high-efficiency visualization and geospatial analysis in climate change research. According to the special features of climate change research (e.g., large volumes of data and heavy computing burden), a tile service-driven architecture is implemented. In this architecture, the original grid dataset is organized into a compressed pyramid tile set so that both the access efficiency and the loading speed for voluminous datasets are improved. Exploiting the tile set and organization of these data models, a gSoap-based web service and a hybrid database file system are configured to maintain a high-efficiency calculation and analysis within the constraints of the hardware resources. We also introduce the open-source virtual globe osgEarth that can generate spatiotemporal dynamics quickly and has high efficiency as well as greater realism. In this way, the proposed architecture can not only provide an intuitive, interaction-friendly data information user interface on the virtual globe but also greatly improve the efficiency of data transmission, three-dimensional visualization and geospatial analysis without a loss in accuracy, which has been proven in our experiments above.

In addition, we combine this architecture to construct a SatCO2





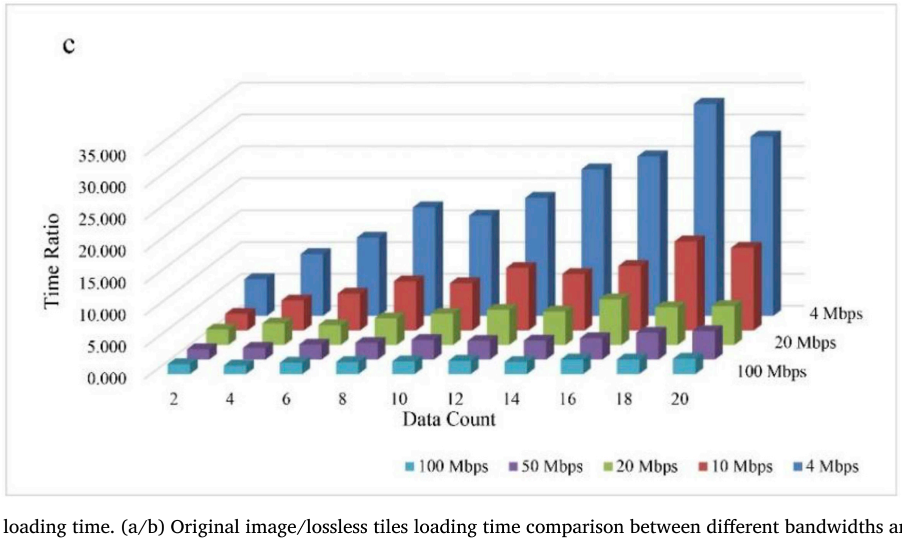


Fig. 10. Comparison of data loading time. (a/b) Original image/lossless tiles loading time comparison between different bandwidths and image counts. (c) Loading time ratio comparison between the original image and the lossless tiles.

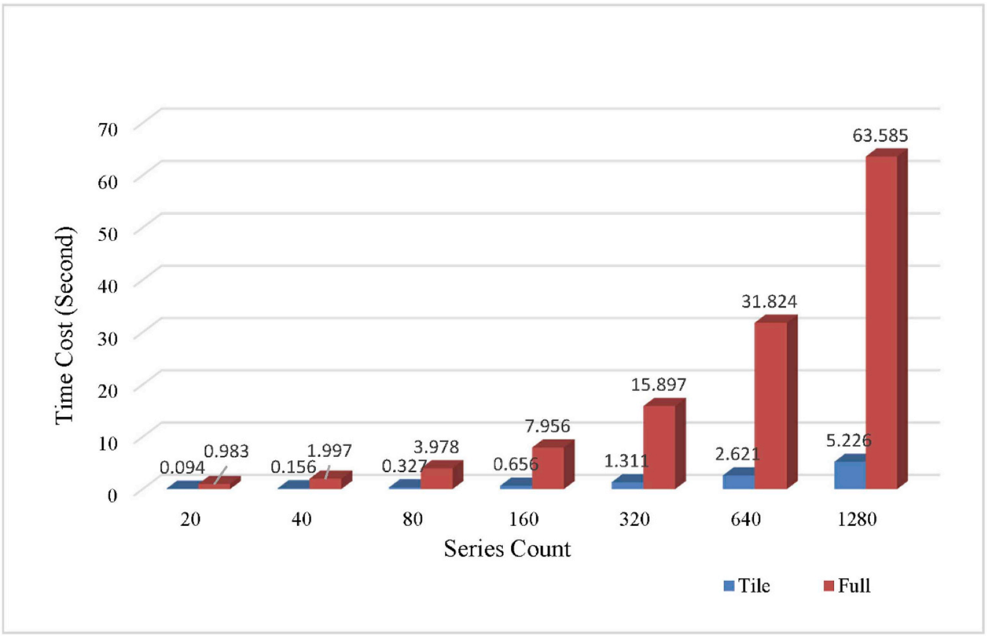


Fig. 11. Time cost comparison of pixel query between the original image and the lossless tiles.

Table 2
RAE of the chlorophyll α concentration from Jan. 2010 to Dec. 2010.

Longitude	113.193	Latitude	18.075				
Date	2010/1/1	2010/1/11	2010/1/21	2010/2/1	2010/2/11	2010/3/1	2010/3/11
Tile Value	0.2014	0.1858	0.1516	0.0960	0.0906	0.1139	0.1419
Real Value	0.2014	0.1858	0.1516	0.0960	0.0906	0.1139	0.1419
Difference	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**
Date	2010/3/21	2010/4/1	2010/4/11	2010/4/21	2010/5/1	2010/5/11	2010/5/21
Tile Value	0.0868	0.0626	0.0795	0.0658	0.0848	0.0924	0.1042
Real Value	0.0868	0.0626	0.0795	0.0658	0.0848	0.0924	0.1042
Difference	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**
Date	2010/6/11	2010/6/21	2010/7/1	2010/7/11	2010/7/21	2010/8/1	2010/8/11
Tile Value	0.0795	0.1003	0.1045	0.1363	0.1104	0.0919	0.1964
Real Value	0.0795	0.1003	0.1045	0.1363	0.1104	0.0919	0.1964
Difference	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Date	2010/8/21	2010/9/1	2010/9/11	2010/9/21	2010/10/11	2010/10/21	2010/11/11
Tile Value	0.1693	0.1254	0.1507	0.1138	0.1305	0.0712	0.1958
Real Value	0.1693	0.1254	0.1507	0.1138	0.1305	0.0712	0.1958
Difference	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**
Date	2010/11/21	2010/12/1	2010/12/11	2010/12/21			
Tile Value	0.1566	0.2541	0.1724	0.1988			
Real Value	0.1566	0.2541	0.1724	0.1988			
Difference	0.0000	0.0000	0.0000	0.0000		**RAE**	**0.0000**

system. By taking full advantage of the architecture, SatCO2 helps researchers to study the ocean carbon cycle and climate change in a virtual globe environment through the Internet even when the volume of data is truly enormous. Compared to Earth Engine, SatCO2 is based on an effective open-source virtual globe and supports high-quality volume rendering, which is suitable for enhancing visualizations. Moreover, since SatCO2 focuses on carbon cycle research, it has a more professional dataset than Earth Engine. However, Earth Engine is web-based and is more convenient for researchers to visualize very large datasets within browsers. Thus, we consider porting the proposed architecture to a web platform for further research.

Although the results presented in this paper demonstrate the successful application of the architecture we proposed, this architecture also has some shortcomings that need to be improved. First, because the original remote sensing image is organized into a pyramid tile set and then stored in HDFS, the tile is broken into smaller sized tiles once

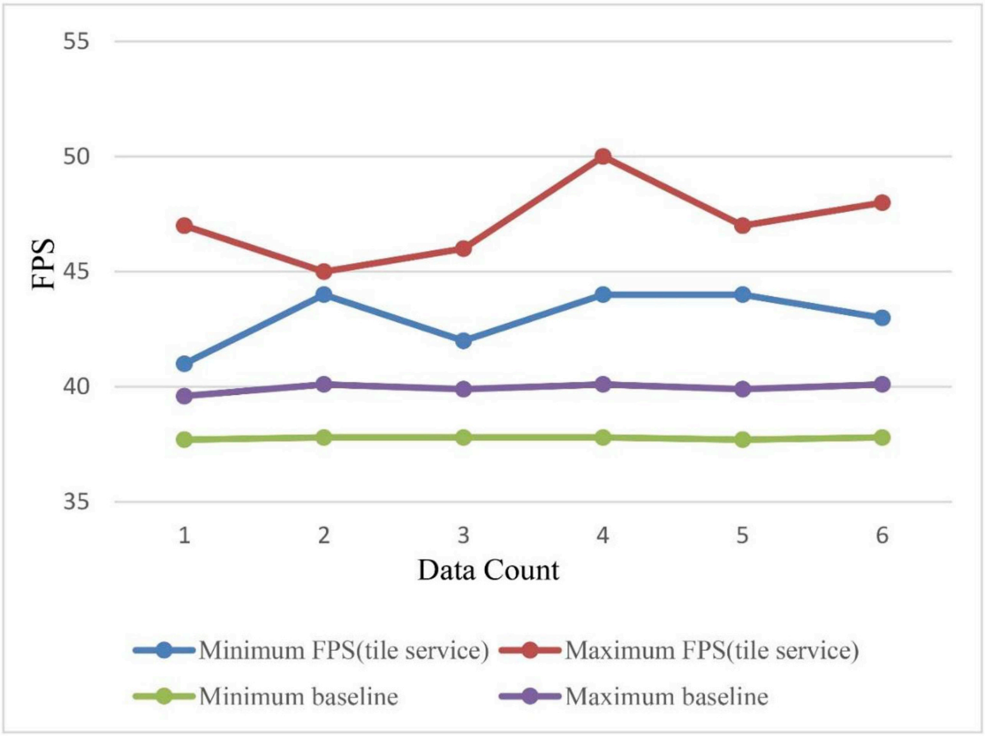


Fig. 12. FPS comparison between whether the tile service strategy is adopted or not.

again and finally stored in different data nodes, which causes a secondary index when performing data loading, undoubtedly reducing efficiency. In future research, we consider combining these two methods, and directly processing the data to lossless tiles during the storing procedure. Second, the proposed lossless tile set strategy is mainly effective for the osgEarth virtual globe and has certain limitations. This strategy will be transferred to other virtual globe platforms for further evaluation.

Acknowledgements

The remote sensing datasets for the long-term marine environment and carbon cycle were provided by the Second Institute of Oceanography (SIO), State Oceanic Administration of China (SOA).

Appendix A. Parameter equations of air-sea CO2 flux calculation

1) Equation of the dissolution efficiency of CO2 (mol/(kg.atm)) (Millero, 1995):

$$\ln(K_H^{CO2}) = -60.2409 + 93.4517 \times (100/T) + 23.3585 \times \ln(T/100) + SSS \times [0.023517 - 0.023656 \times (T/100) + 0.0047036 \times (T/100)^2] \qquad (A1)$$

$$T = SST/°C + 273.15 \qquad (A2)$$

2) Sea water density can be calculated by the function of surface water temperature and salinity (Millero and Poisson, 1981).

$$\rho = ?\rho_w + A*SSS + B*SSS^{3/2} + C*SSS^2?*10^{-3} \qquad (A3)$$

where:

$$A = 0.824493 - 4.0899*10 - 3*SST + 7.6438*10 - 5*SST^2 - 8.2467*10 - 7*SST^3 + 5.3875*10 - 9*SST^4 \qquad (A4)$$

$$B = -5.72466*10^{-3} + 1.0227*10^{-4}*SST - 1.6546*10^{-6}*SST^2 \qquad (A5)$$

$$C = 4.8314*10^{-4} \qquad (A6)$$

$$\rho_w = 999.842594 + 6.793952*10^{-2}*SST - 9.09529*10^{-3}*SST^2 + 1.001685*10^{-4}*SST^3 - 1.120083*10^{-6}*SST^4 + 6.536332*10^{-9}*SST^5 \qquad (A7)$$

3) To calculate the monthly average flux, it is often necessary to consider the influence of high-frequency wind speed change (e.g, daily) on the monthly average wind speed, using a coefficient of C2 (Wanninkhof et al., 2002). The C2 coefficient is not needed when calculating daily flux.

$$C_2 = \frac{(U_j^2)_{mean}}{(U_{mean})^2} \qquad (A8)$$

U_j is high-frequency satellite-derived wind speed (e.g., daily), and U_{mean} is satellite-derived monthly average wind speed, both with units of m/s.

4) Based on the relationship between the gas transfer velocity (k) and the wind speed at 10 m above sea level (U_{10}), the commonly used equations for calculating k are shown in the table below.

Table A1
Commonly used equations for calculating gas transfer velocity.

	Equation	References
1	$k_{660} = 0.31 \times U_{10}^2$(Instantaneous wind speed)	Wanninkhof (1992)
	$k_{660} = 0.39 \times U_{10}^2$ (Long-term average wind speed)	
2	$k_{660} = 0.27 \times U_{10}^2$	Sweeney et al. (2007)
3	$k_{600} = 0.266 \times U_{10}^2$	Ho et al. (2006)
4	$k_{660} = 0.0283 \times U_{10}^3$	Wanninkhof and McGillis (1999)

k_{660} and k_{600} indicate the k with Schmidt numbers (Sc) of 660 and 600, respectively.

$$\mathrm{k}=k_{600} * (Sc/600)^{-0.5} \tag{A9}$$

$$\mathrm{k}=k_{660} * (Sc/660)^{-0.5} \tag{A10}$$

When the sea surface temperature ranges from 0 to 30 °C, the Sc can be calculated with the following equation:

$$\mathrm{Sc} = 2073.1 - 125.62 \times \mathrm{SST} + 3.6276 \times \mathrm{SST}^2 - 0.043219 \times \mathrm{SST}^3 \tag{A11}$$

For the calculation of the satellite-derived air-sea CO2 flux in the Open Ocean, the k-U_{10} equations of #1 (long-term wind speed) and #2 in the above table are commonly used.

Appendix B. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.envsoft.2019.04.005.

Funding

This research was supported by the National Key Research and Development Program of China [2018YFB0505000, 2016YFC1400903], the National Natural Science Foundation of China [41871287, 41671391], and the Public Science and Technology Research Funds Projects for Ocean Research [201505003].

References

Adelson, Edward H., Anderson, Charles H., Bergen, James R., Burt, Peter J., Ogden, Joan M., 1984. Pyramid methods in image processing. RCA Eng. 29 (6), 33–41.
Arthur, Dan K., Lasher-Trapp, Sonia, Abdel-Haleem, Ayman, Klosterman, Nicholas, Ebert, David S., 2010. A new three-dimensional visualization system for combining aircraft and radar data and its application to RICO observations. J. Atmos. Ocean. Technol. 27 (5), 811–828.
Borthakur, D., 2008. HDFS architecture guide. Hadoop Apache Proj. 53, 1–13.
Chang, George, Malhotra, Shan, Paul, Wolgast, 2011. Leveraging the cloud for robust and efficient lunar image processing. In: In Aerospace Conference, 2011 IEEE. IEEE, pp. 1–8.
Chen, Shupeng, van Genderen, John, 2008. Digital Earth in support of global change research. Int. J. Digit. Earth 1 (1), 43–65.
Cheng, R., Chen, J., Cao, M., 2019. A virtual globe-based three-dimensional dynamic visualization method for gas diffusion. Environ. Model. Softw 111, 13–23.
Darles, Emmanuelle, Crespin, Benôıt, Ghazanfarpour, Djamchid, Gonzato, Jean-Christophe, 2011. A survey of ocean simulation and rendering techniques in computer graphics. Comput. Graph. Forum 30, 43–60 (Wiley Online Library).
David, Salomon, 2004. Data Compression: the Complete Reference. Springer Science & Business Media.
Deutsch, Peter, 1996. DEFLATE Compressed Data Format Specification Version 1.3. No. RFC 1951.
Drebin, Robert A., Carpenter, Loren, Hanrahan, Pat, 1988. Volume rendering. In: ACM Siggraph Computer Graphics, vol 22. ACM, pp. 65–74.
Du, Zhenhong, Fang, Lei, Bai, Yan, Zhang, Feng, Liu, Renyi, 2015. "Spatio-temporal visualization of air–sea CO2 flux and carbon budget using volume rendering. Comput. Geosci. 77, 77–86.
Foo, Soo Mee, Lee, Wei Meng, 2002. Simple object access protocol (SOAP) and web services. In: XML Programming Using the Microsoft XML Parser. Springer, pp. 347–391.
Fortner, Brand, 1998. HDF: the hierarchical data format. Dr. Dobb's J. Softw. Tools Prof. Program. 23 (5), 42.
Gong, Peng, Zhao, Y.C., Liang, Yu, Lu, Liang, 2011. Development of an integrated software platform for global mapping and analysis. Geomatics World 2, 34–37.
Gorelick, Noel, Hancher, Matt, Dixon, Mike, Simon, Ilyushchenko, Thau, David, Moore, Rebecca, 2017. Google earth engine: planetary-scale geospatial analysis for everyone. Rem. Sens. Environ. 202, 18–27.
Gould, Michael, Craglia, Max, Goodchild, Michael F., Annoni, Alessandro, Camara, Gilberto, Kuhn, Werner, Mark, David, et al., 2008. Next-generation Digital Earth: A Position Paper from the Vespucci Initiative for the Advancement of Geographic Information Science.
Green, Simon, 2008. Volumetric Particle Shadows. NVIDIA Developer Zone.
Hunter, Gregory M., Steiglitz, Kenneth, 1979. Operations on images using quad trees. IEEE Trans. Pattern Anal. Mach. Intell. (2), 145–153.
Ho, D.T., Law, C.S., Smith, M.J., et al., 2006. Measurements of air-sea gas exchange at high wind speeds in the Southern Ocean: implications for global parameterizations. Geophys. Res. Lett. 33 (16).
Kang, Sinkyu, et al., 2014. Effects of spatial and temporal climatic variability on terrestrial carbon and water fluxes in the Pacific Northwest, USA. Environ. Model. Softw 51, 228–239.
Laur, David, Hanrahan, Pat, 1991. Hierarchical splatting: a progressive refinement algorithm for volume rendering. In: ACM SIGGRAPH Computer Graphics, vol 25. ACM, pp. 285–288.
Lee, Yeonhee, Lee, Youngseok, 2013. Toward scalable internet traffic measurement and analysis with hadoop. Comput. Commun. Rev. 43 (1), 5–13.
Li, Jing, Wu, Huayi, Yang, Chaowei, Wong, David W., Xie, Jibo, 2011. Visualizing dynamic geosciences phenomena using an octree-based view-dependent LOD strategy within virtual globes. Comput. Geosci. 37 (9), 1295–1302.
Li, Wenwen, Wang, Sizhe, 2017. PolarGlobe: a web-wide virtual globe system for visualizing multidimensional, time-varying, big climate data. Int. J. Geogr. Inf. Sci. 31 (8), 1562–1582.
Liang, Jianming, Gong, Jianhua, Li, Wenhang, Ibrahim, Abdoul Nasser, 2014. Visualizing 3D atmospheric data with spherical volume texture on virtual globes. Comput. Geosci. 68, 81–91.
Liang, Jianming, Gong, Jianhua, Zhou, Jieping, Ibrahim, Abdoul Nasser, Li, Ming, 2015. An open-source 3D solar radiation model integrated with a 3D Geographic Information System. Environ. Model. Softw 64, 94–101.
Liu, Po, Gong, Jianhua, Yu, Miao, 2015a. Graphics processing unit-based dynamic volume rendering for typhoons on a virtual globe. Int. J. Digit. Earth 8 (6), 431–450.
Liu, Po, Gong, Jianhua, Yu, Miao, 2015b. Visualizing and analyzing dynamic meteorological data with virtual globes: a case study of tropical cyclones. Environ. Model. Softw 64, 80–93.
Loveland, Thomas R., Dwyer, John L., 2012. Landsat: building a strong future. Rem. Sens. Environ. 122, 22–29.
Millero, F.J., 1995. Thermodynamics of the carbon dioxide system in the oceans. Geochem. Cosmochim. Acta 59 (4), 661–677.
Millero, F.J., Poisson, A., 1981. International one-atmosphere equation of state of seawater. Deep Sea Res. Part A. Oceanogr. Res. Pap. 28 (6), 625–629.
Monte, Cristian Federico Perez, Piccoli, Fabiana, Luciano, Cristian, Rizzi, Silvio, Bianchini, Germ´an, Caymes Scutari, Paola, 2013. Estimation of volume rendering efficiency with gpu in a parallel distributed environment. Procedia Comput. Sci. 18, 1402–1411.
Myers, Samuel S., Patz, Jonathan A., 2009. Emerging threats to human health from global environmental change. Annu. Rev. Environ. Resour. 34, 223–252.

NASA, 2018NASA. n.d,. Climate Change: how Do We Know? n.d. https://climate.nasa.gov/evidence/, Accessed date: 16 March 2018.

Nemani, R., Votava, P., Michaelis, A., Melton, F., Milesi, C., 2011. Collaborative supercomputing for global change science. Eos, Trans. Am. Geophys. Union 92 (13), 109–110.

Nvidia, C., 2010. Programming guide[J].

Open C V, 2013. OpenCV documentation[J].

Organization, World Health, et al., 2008. Climate Change and Health.

Powell, Mark W., Rossi, Ryan A., Shams, Khawaja, 2010. A scalable image processing framework for gigapixel mars and other celestial body images. In: In Aerospace Conference, 2010 IEEE. IEEE, pp. 1–11.

Soille, P., Burger, A., De Marchi, D., Kempeneers, P., Rodriguez, D., Syrris, V., Vasilev, V., 2018. A versatile data-intensive computing platform for information retrieval from big geospatial data. Future Gener. Comput. Syst. 81, 30–40.

Song, Yuyan, Ye, Jing, Svakhine, Nikolai, Lasher-Trapp, Sonia, Baldwin, Mike, Ebert, David, 2006. An atmospheric visual analysis and exploration system. IEEE Trans. Vis. Comput. Graph. 12 (5), 1157–1164.

Standart, Gordon D., et al., 2011. Geospatial visualization of global satellite images with Vis-EROS. Environ. Model. Softw 26 (7), 980–982.

Sweeney, C., Gloor, E., Jacobson, A.R., et al., 2007. Constraining global air-sea gas exchange for CO2 with recent bomb 14C measurements. Glob. Biogeochem. Cycles 21 (2).

Vasco, D.W., Rucci, Alessio, Ferretti, Alessandro, Novali, Fabrizio, Bissell, R.C., Ringrose, P.S., Mathieson, A.S., Wright, I.W., 2010. Satellite-based measurements of surface deformation reveal fluid flow associated with the geological storage of carbon dioxide. Geophys. Res. Lett. 37 (3).

Wanninkhof, R., Doney, S.C., Takahashi, T., et al., 2002. The effect of using time-averaged winds on regional air-sea CO ∼ 2 fluxes. Geophys. Monogr.-Am. Geophys. Union 127, 351–356.

Wanninkhof, R., McGillis, W.R., 1999. A cubic relationship between air-sea CO2 exchange and wind speed. Geophys. Res. Lett. 26 (13), 1889–1892.

Wanninkhof, R., 1992. Relationship between wind speed and gas exchange over the ocean. J. Geophys. Res.: Oceans 97 (C5), 7373–7382.

Wang, Y., Huynh, G., Williamson, C., 2013. Integration of Google Maps/Earth with microscale meteorology models and data visualization. Comput. Geosci. 61, 23–31.

Woodcock, Curtis E., Allen, Richard, Anderson, Martha, Belward, Alan, Bindschadler, Robert, Cohen, Warren, Gao, Feng, et al., 2008. Free access to Landsat imagery. Science 320 (5879) 1011–1011.

Yu, Le, Gong, Peng, 2012. Google Earth as a virtual globe tool for Earth science applications at the global scale: progress and perspectives. Int. J. Remote Sens. 33 (12), 3966–3986.

Zhang, Tong, Jing, Li, Liu, Qing, Huang, Qunying, 2016. A cloud-enabled remote visualization tool for time-varying climate data analytics. Environ. Model. Softw 75, 513–518.